Information-Rich Virtual Environments: Challenges and Outlook

NASA Virtual Iron Bird Workshop, 2004

Nicholas F. Polys Doug A. Bowman Chris North
Department of Computer Science
Center for Human-Computer Interaction
Virginia Polytechnic Institute and State University

Abstract

Increasingly designers, engineers, and scientists require 'Integrated Information Spaces' where spatial, abstract, and temporal data are simultaneously available and linked. To address this problem in our work we are developing Information-Rich Virtual Environments (IRVEs). An IRVE combines the capabilities of *virtual environments* and *information visualization* to support the integrated exploration of spatial, abstract, and temporal data This paper enumerates the display and interaction requirements for Integrated Information Spaces and describes our first attempts to meet them in the context of international standards such as XML, VRML, and X3D. We will describe our IRVE designs and implementations from a user-centered, usability engineering perspective and detail their strengths and weaknesses for publishing and service architectures. We have prototyped systems to unify heterogeneous information types for applications in cheminformatics and biomedical simulation. Current standards promise to enable better IRVEs including X3D's support for Metadata and the proposed components for Annotation and Compositing. Finally, we examine how our work can inform the standards design process to insure that crucial IRVE functionality can be supported.

1.0 Introduction

Across a wide variety of domains, analysts and designers are faced with complex systems that include spatial objects, their attributes and their dynamic behaviors. In order to study and understand these systems, users need a unified platform to explore the complex relationships between their heterogeneous data types. Next generation digital tools must address this need for integration of spatial, abstract, and temporal information; we have described this as the goal of 'Information-Rich Virtual Environments' [Bowman et al, 2003].

Consider an engineer, 'E', working on a complex aerospace craft such as the Space Shuttle. The craft is aging and the tolerances of the original parts are suspect. The engineer is tasked with designing a new gear assembly for the tailfin brake. E builds a 3D geometric model of the assembly in a CAD program, specifying its dimensions and the materials for its construction. E must then test the assembly design and its parts for physical tolerances using a meshing program and finite-element simulator. E must specify the kinetic forces of the assembly such as gears, locks, fulcri etc. After the simulator is run, E analyzes the results looking for weak points and insuring that all parts are within physical requirements. E repeats this process a number of times to satisfy the specified design constraints of material's weights and stress limits. When he is satisfied, he saves the candidate model and simulation results into a database that represents the craft in its entirety. But E is not done; he must also confirm how his design affects the whole craft in flight and damage scenarios. E's new part is then evaluated in the context of the other shuttle systems. Each scenario is linked to prioritized causal chains in a knowledgebase that infers mission consequences due to the particular flight/damage scenario.

How many applications did E use? How many times was he required to switch between a spatial view (e.g. a design application), a temporal view (e.g. a simulator application), and an abstract view (e.g. a information visualization application)? Was any model data lost or added between the applications? Was he able to view the impacts of his design changes simultaneously within one environment? On the same machine? This scenario illustrates the problem of integrated information spaces- current data models, applications, and presentations are fragmented and inefficient for users.

Virtual environments (VEs) excel at providing users a greater comprehension of spatial objects, their perceptual properties and their relations. Perceptual information includes 3D spaces that represent physical or virtual objects and phenomena including geometry, lighting, colors, and textures. As users navigate within a complex VE model, they may need access to information related to the world and objects in the space (such as name, function, attributes, etc.). This is the domain of Information Visualization, which is concerned with improving how users perceive, understand, and interact with visual representations of abstract information [Card et al, 1999]. This abstract (or symbolic) information could include text, links, numbers, graphical plots, and audio/video annotations. Both perceptual and abstract information may change over time reflecting its temporal aspects. Unfortunately, most systems do not allow users flexible exploration and examination of dynamic abstract information in conjunction with a dynamic VE.

We believe that this integrated capability is necessary for users to gain a full understanding of complex relationships in their heterogeneous data, and to create usable research, design, and decision-support applications. *Information-rich virtual environments* (IRVEs) start with realistic perceptual information and enhance it with related abstract information. The methods of VEs and Info Vis are combined in a concerted way to enable a unified approach to exploring the space and information. We believe it is essential that IRVE can integrate VE softwares and InfoVis software for the qualities of parallelism of display and coordination of views.

Our hypothesis is that the IRVE strategy can give users easier, integrated access to information and help them to generate insight through stronger mental associations between physical, abstract, and temporal information while preserving mental models of each type of information. The primary goal of our work is to build an effective user interface framework for information-rich databases that:

- Provides integrated visualization of **spatial**, **abstract**, and **temporal** data
- Supports interactive navigation across multiple levels of spatial, abstract and temporal scales
- Enables insight generation and decision making from complex, multi-scale relationships in heterogeneous data
- Feels familiar to users with maximum similarity to the domain area in which they operate, by relating the visualizations to actual physical structures
- Is intuitive and usable after minimal training

In this paper, we will look at the design requirements for next-generation IRVE software tools and describe our standards-based prototype implementations. We will reflect on emerging technologies and how our early work can inform, improve, and enable effective IRVE platforms.

2.0 Related Work

Munro et al [2002] outlined the cognitive processing issues in virtual environments by the type of information they convey (Table 1). In reviewing VE presentations and tutoring systems, the authors note that VEs are especially appropriate for: navigation and locomotion in complex environments, manipulation of complex objects ad devices in 3D space, learning abstract concepts with spatial characteristics, complex data analysis, and decision making. IRVEs are especially concerned with improving the latter three.

Location Knowledge	**Structural Knowledge**
• Relative position • Navigation • 'How to view' (an object) • 'How to use' (an object access & manipulation affordances e.g. a door)	• Part-whole • Support-depend (i.e. gravity) • Containment
Behavioral Knowledge	**Procedural Knowledge**
• Cause-and-effect • Function • Systemic behavior	• Task prerequisite • Goal hierarchy • Action sequence

Table 1: Taxonomy of knowledge types for VE presentations (per Munro et al 2002).

IRVEs are relatively new and the effectiveness of various information and interaction designs are not yet known. Nonetheless, we believe giving users a unified interactive experience of some process or phenomenon can improve learning, performance, and the accuracy of their mental models. While IRVEs provide the potential for users to integrate heterogeneous information from diverse sources in one session, most IRVEs are simplistic consisting of worlds with animations and labels or linked applications in external windows. Few exploit the full design space to give users flexible control over their views and interactions, but promising work is emerging in academia and industry that deserves mention.

2.1 IRVEs in Education & Training

Parallel Graphics' Virtual Manuals solution demonstrates the integration of abstract information (e.g. names, part numbers) within the spatial world and with external windows for training applications in operation and maintenance. Temporal information is rendered through animated sequences of assembly and dis-assembly for example. This approach is consistent with HCI research in comprehension and user's mental models. For example, users gained improved situational awareness, understanding, and recall through multimedia presentations integrating these features [Sutcliffe & Faraday, 1994; Faraday & Sutcliffe, 1996].

Distance learning researchers [NDSU WWWIC, 2004] have shown improved student performance by augmenting science lectures with desktop virtual environments including The Virtual Cell environment for biology and the processes of cellular respiration [McClean et al 2002; White et al 1999; Saini-Eidukat et al, 1999]. The Virtual Cell courseware uses VRML to display and label cell structures and agents and to animate their processes and interactions. Early versions used the EAI and Java to track student's progress and current versions use the Xj3D toolkit.

The ScienceSpace Project showed that conceptual learning can be aided by features of immersive VEs such as: their spatial, 3-dimensional aspect, their support for users to change their frames of reference, and the inclusion of multi-sensory cues [Salzman et al, 1999]. The curriculum modules included learning about dynamics and interactions for the physics of Newton, Maxwell, and Pauling. These environments provide menus, integrated object labels, and field lines to unify perceptual and abstract data. Given the results above, it seems likely that this advantage would also transfer to desktop courseware and applications.

2.2 IRVEs in 4D Design

The trend toward IRVEs can also be seen in Sheppard's recent survey [2004] of construction-related 4D applications. Applications such as PM-Vision and ConstructSim for example, provide an integrated workspace for construction project managers to relate various costs and timelines to their models. Users can switch back and forth between views to examine and change parameters and scenarios. In addition, Domos Software's 4D Builder Suite, integrates CAD geometries and project planning softwares to give planners an integrated view of various material and scheduling choices; the suite uses XML to describe the relations between VRML object identifiers and the planning and profile definitions [Domos, 2004]. These applications are beginning to demonstrate the power of IRVEs to improve efficiency by minimizing potential conflicts and costs in the planning stage.

The Industry Alliance for Interoperability [IAI, 2002] has specified a data model for AEC applications known as Industry Foundation Classes (IFC). The IFC schema organizes objects, characteristics, and inter-relationships for project management and interchange. The class hierarchy integrates materials, costs, schedules, and organizations in order that project information can be shared across applications and kept consistent. The data model specified in the IFC provides a proven approach to management of architectural and engineering related data that is highly relevant for engineering IRVE backends.

3.0 Challenges and Requirements

To design and deploy integrated information spaces that meet user requirements, developers face a number of challenges. Across application domains, these requirements involve display and interaction techniques to relate heterogeneous data types to one another and to find spatial patterns and trends in the data. To date, interface features and software architectures have been ad hoc and primarily domain-specific. In order to support IRVE functionality, a rational design approach is required. The data and presentation problems experienced by our engineer E are common to simulation and design applications in a number of domains.

Constructing optimal IRVEs will require development of new information access interfaces, efficient database storage and integration, and open software architecture for mapping data to real-time graphical displays. From the end-user perspective, the IRVE must be intuitive and easy to use, facilitate insight generation from massive and complex databases, drive data retrieval, and support perceptual similarity to the domain area. Iterative participatory design and usability engineering with user populations can ensure that their needs are met [Rosson & Carroll, 2002].

3.1 User Centered Design

User-centered design refers to a product design process that considers the user population and its demographics, requirements, work environment, and tasks as the primary driving force. Many researchers have shown that a user-centered design process produces systems that are more usable, more accepted, and less problematic than systems designed with the focus on features. User-centered design is a part of the overall process known as usability engineering, which is based on an iterative design-evaluate-redesign cycle.

Usability engineering is an approach to software development in which target levels of system usability are specified in advance, and the system is engineered toward these measures. Designers work through a process of identifying the user **activities** the system must support, the **information** that is required for users to understand the system and task state, and the **interactions** required to support those tasks. For each feature in a design, usability engineers identify the tradeoffs of that feature and then analyze their claims that the feature resolves the tradeoff. Usability evaluation of VEs presents issues above and beyond 2D interfaces. For a summary, see Gabbard et al. [1999] and Bowman et al. [2002].

3.1.1 IRVE Activity Design

Because objects and object kinds may carry a range of attribute types, depth of attribute detail, and inter-relations, the multi-scale integration of heterogeneous data sets and knowledge bases is of prime importance to usable IRVE interfaces. As users navigate throughout an environment and across scales, they need dynamic access to object properties and parameters. Figure 1 summarizes the requirements for multi-scale IRVE navigation and shows that users may not only need access to one scale of environment, but also overview and detail information about systems and subsystems.

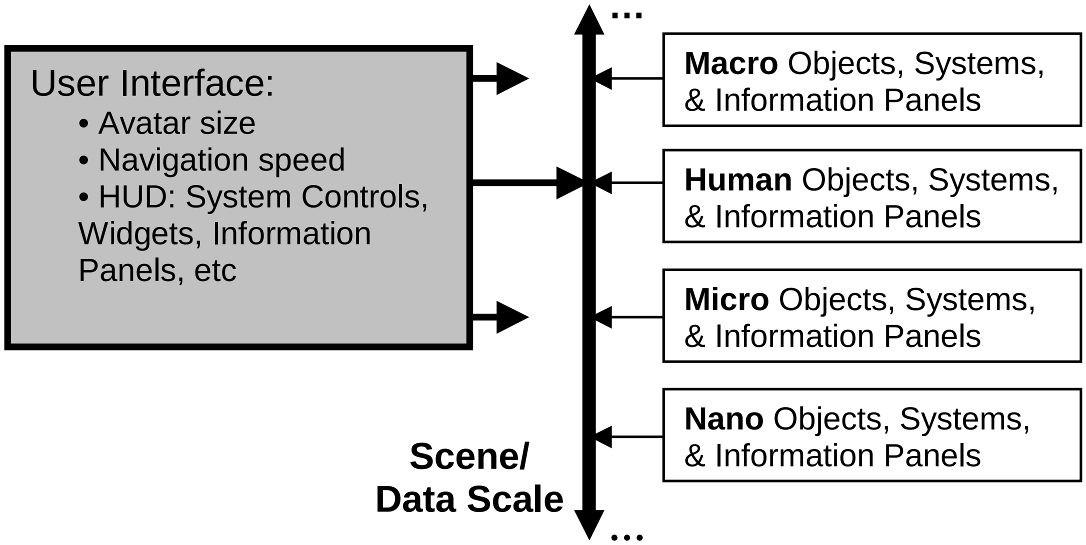

Figure 1: Spatial and Abstract Scale requirements for IRVE Activities

One effective approach to the multi-scale navigation of information is summarized in Shneiderman's mantra of information design: "Overview first, zoom and filter, then details-on-demand" [1996]. In addition, Bederson et al [1996] proposed that interface designers appeal to user's knowledge about the real world, i.e. that objects appear and behave differently depending on the scale of the view and the context. They have termed this 'interface physics' and demonstrated the Pad++ system for 'semantic zooming' where both the content of the representation and the manipulation affordances it provides are directly and naturally available to the user.

Due to the rich nature of IRVEs, it is important to consider user task and resource analysis to determine a 'task-knowledge structure' and formalize it as an entity-relationship model (i.e. Sutcliffe & Faraday, 1994). This model enables the effective design of multimedia interfaces and presentation scripting- e.g. what media resources the user needs visual access to when. This is an important consideration for IRVE design as it intends to formally identify items that need user attention and minimize perceptual overload and interference during task performance.

3.1.2 IRVE Information Design

The IRVE information design problem focuses on the representation and layout of embedded abstract information in perceptually realistic virtual environments. There are many possible representations for a given dataset, such as a table, a plot or graph, a text description, or an equation. Placing this information (layout) within the 3D VE is not a trivial issue - there are a number of possibilities as to how abstract information is related to the perceptual information in an IRVE [Bowman et al, 2003]. The displayed information needs to be visible for the user; it should not occlude important objects in the world; and it should not interfere with other displayed information. We will also develop techniques for managing the layout of information within a VE.

Our previous work in IRVEs has focused on simple embedded abstract information types such as text, audio, and images. We found that embedded information was most effective and useful when it was "tightly coupled" to the surrounding sensory environment through spatial location, visual references, or "spatial hyperlinks" [Bowman et al, 1999]. We have also been investigating a wide range of text and graphical layout schemes in IRVEs. Our current software architecture provides developers with many different options for the "layout" of text objects, such as heads-up displays (HUDs), embedded text labels fixed to objects in the 3D environment, embedded labels that automatically position themselves to maintain legibility, and text with varying levels of detail depending on the context. This architecture allows us to quickly prototype different information display techniques, and to run experiments comparing two or more techniques.

Our theoretical framework includes a notion of the design space for IRVE information display techniques. In other words, what are the possible ways that abstract information should be included within a VE? This includes issues such as the type of abstract information (e.g. text, audio, graphics), where the information should appear (e.g. in a heads-up display, in a spatial location attached to an object), how links among information should be represented, and so on. Our proposed design matrix is shown in Table 2 below.

Abstract information design parameter	Psychological process	Usability impact
Visual attributes: - color - fonts - size - background - transparency	**Perception**	- Legibility - Readability - Occlusion
Layout attributes: - location - association - density	**Recognition**	- Display-, world-, user-, object-fixed - Implicit and explicit, visual and spatial - Occlusion
Aggregation: - level of information encoding - type of visualization	**Integration**	- Conceptual categories & abstractions
	Comprehension & Cognition	- Satisfaction - Effectiveness (*i.e.*, for decision making and insight)

Table 2: IRVE design matrix for abstract information display

3.2.3 IRVE Interaction Design

Our prior IRVE research emphasized the design of usable interaction techniques for 3D navigation, information access, and information filtering (Bowman et al, 1998; Bowman et al, 1999). We found that giving users the ability to navigate in multiple ways was crucial. Users may want to navigate the spatial environment or the abstract information only, or they may wish to navigate spatially based on the content of the abstract information, or they may wish to view abstract information based on their spatial location. All of these must be supported. We also found that users want precise but simple control over the types of information displayed in the environment (filtering).

From an IRVE interaction design perspective, users may index into the perceptual data through the abstract data or into the abstract data through the spatial. Accessing information can be considered a special case of two more general VE interaction tasks: object selection and system control. First, information associated with an object can be retrieved by selecting that object. Many techniques have been developed for 3D selection, such as ray-casting [Mine, 1995] and Go-Go [Poupyrev et al, 1996]; for a summary, see [Bowman, 2002]. Second, users may want to perform actions on selected objects or in a system-wide manner. This implies the use of system control techniques such as virtual menus, voice commands, or text input.

Finally, we will work to define the design space for interaction techniques in IRVEs. Since many of the basic user tasks in IRVEs will be the same as those in general VEs, such as navigation, object selection and manipulation, and system control, we can use existing technique classifications for those tasks [Bowman et al, 2001]. However, some other user tasks (such as information query) will be common enough in IRVEs that they should be considered separately. These include the facilities for:

- *Overview*: data loading, choosing the data and display method
- *Details-on-demand*: queries, lenses, zooming via 3D navigation, pop-ups, spatial hyperlinks
- *Relating Perceptual and Abstract Information*: rendering associations; brushing and linking
- *Recognizing Patterns and Trends*: comparing and integrating multiple IRVE information sources

Coordinating 2D and 3D views in a virtual environment not only requires the sharing of addressable data objects and event communication across the applications, but also special interaction facilities for immersive displays such as an HMD or CAVE (e.g. [Polys et al, 2004a]). For example, a multiple views approach on a desktop can use mouse events. User input in immersive systems is typically through the use of a 6 degree-of-freedom tracked ‘wand’ or stylus, which have a number of buttons. In addition, the wand provides a thumb-size joystick. Using such tracked input devices, users may need interaction facilities to index into the spatial, abstract, and temporal information based on any of the other information types. A recent experiment from our group examined exploration and search tasks using a variety of layout and association techniques in immersive environments and demonstrated a significant advantage to a combination of HUD displays and Go-Go navigation [Chen et al, 2004]

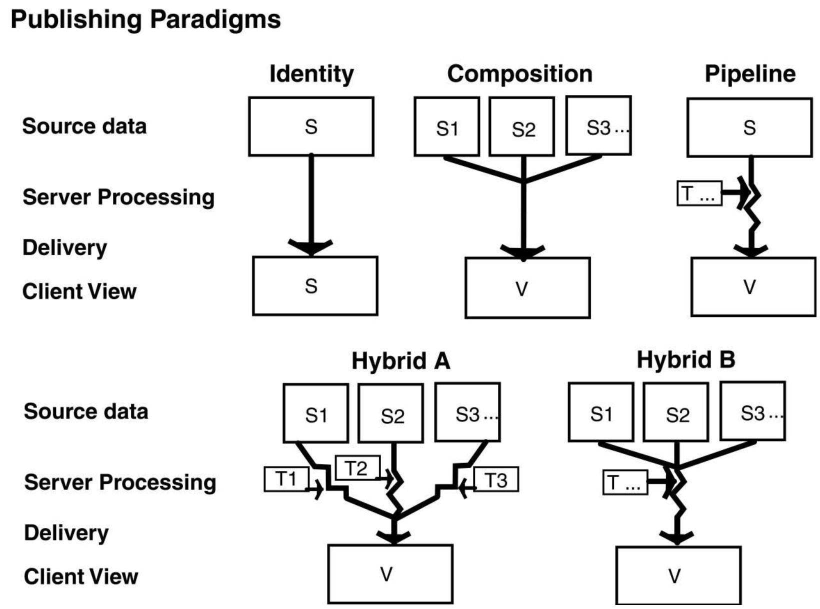

Figure 3: Source (S), View (V), and Transformation (T) schemes for information publishing (from Polys 2004c)

3.2 Software Architectures

The typical pipeline for information visualization was described in Card et al. [1999] and shown in Figure 2. Beginning with raw data, which may be highly dimensional and heterogeneous, data transformations are applied to produce a ‘Data Table’ that encapsulates the records and attributes of interest. The data table also includes metadata which describes the respective axis of the data values. Visual mappings are then applied to the data table to produce the visual structures of the visualization. The final transformation stage involves defining the user views and navigational mechanisms to these visual structures. As the user interactively explores the data, this process is repeated. Ideally, the user has interactive control (feedback) on any step in the process (the data transformation, visual mappings, and view transformations).

Delivering IRVE content to users can be a formidable challenge. As data models and delivery technologies continue to evolve, developers must consider efficient architectures for integrating a variety of data and knowledge bases and composing content. Application servers and Web services can be characterized by the way they transform content and compose a user view. Publishing paradigms are summarized in Figure 3. One item of note is that client-side applications may also perform transformations.

4.0 Prototype IRVE Applications

With our initial visualization software designs, we have explored IRVE tasks and scenarios that must be supported. These concern how well users can locate, relate, and understand information across the spatial and abstract visualizations. Here we summarize the usability evaluations of two early approaches to IRVE design: linked views [Polys et al, 2004a], and embedded visualization [Polys et al, 2004b]. From a software development perspective, maintaining unique data identifiers across transformations and between

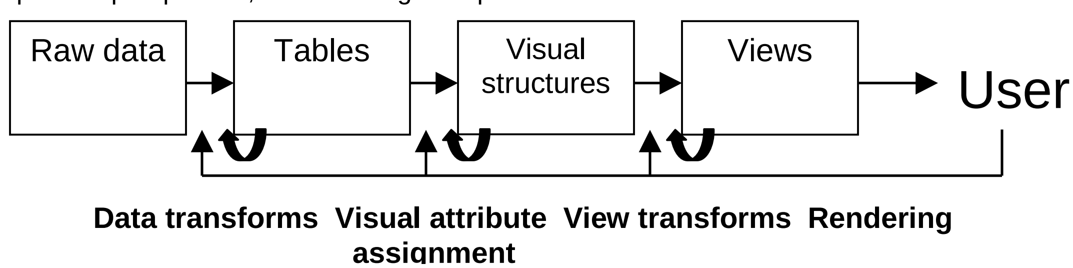

Figure 2: Processing in a typical visualization pipeline (from Card et al, 1999)

applications and using simple interface events to connect them seems sufficient to provide complex multiple-view functionality of heterogeneous data sets and simulation results.

4.1 Cheminformatics: Snap2Diverse

Our molecular visualization prototype (Figures 4 and 5) examines linking to external visualization tools within the context of an IRVE (Snap2Diverse) [Polys et al, 2004a]. We wanted to identify usability issues when using multiple linked views of heterogeneous physical and quantitative data in an IRVE. While this framework for coordinated viewing is applicable to a variety of data domains such as medicine, engineering, and architecture, here we use a set of Chemical Markup Language (CML) files since it exemplifies the combination of spatial and abstract information types.

The prototype included an immersive 3D display of the physical structures of several molecules (including caffeine and histamine). In addition, several linked data visualizations of other quantitative and textual molecular properties were embedded in the environment as a 'hanging picture'. The contents of the hanging picture is an application called Snap-Together Visualization [North & Schneiderman, 2000; North, 2001; North et al, 2002]. In Snap, users can interactively examine relationships in their data by using tightly coupled multiple views, selecting and loading records of interest. In Snap, simple coordination events and record IDs provide the support for complex visualization functionality. We extended Snap to use DIVERSE [Kelso et al, 2002] as a visualization component. The architecture for the system is shown in Figure 6. For example, selecting the strongest bonds from a 2D scatterplot would select the spatial rendering of those bonds in the structure.

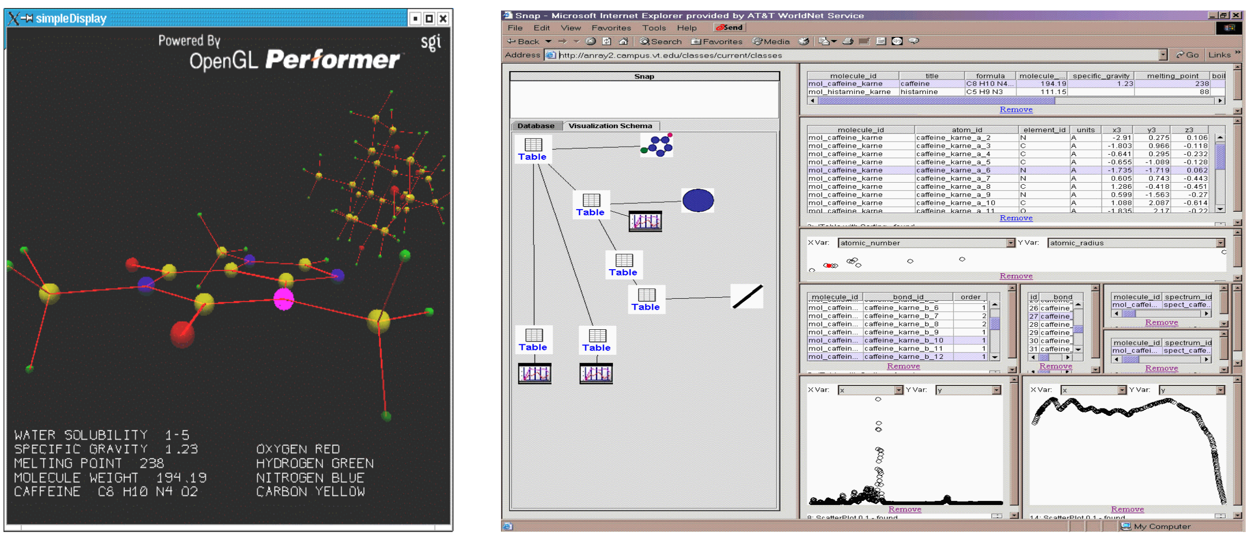

Figure 4: Coordinated VE (showing spatial molecular structure) and information visualization (showing quantitative data about structure)– an example of the multiple-views approach on a desktop.

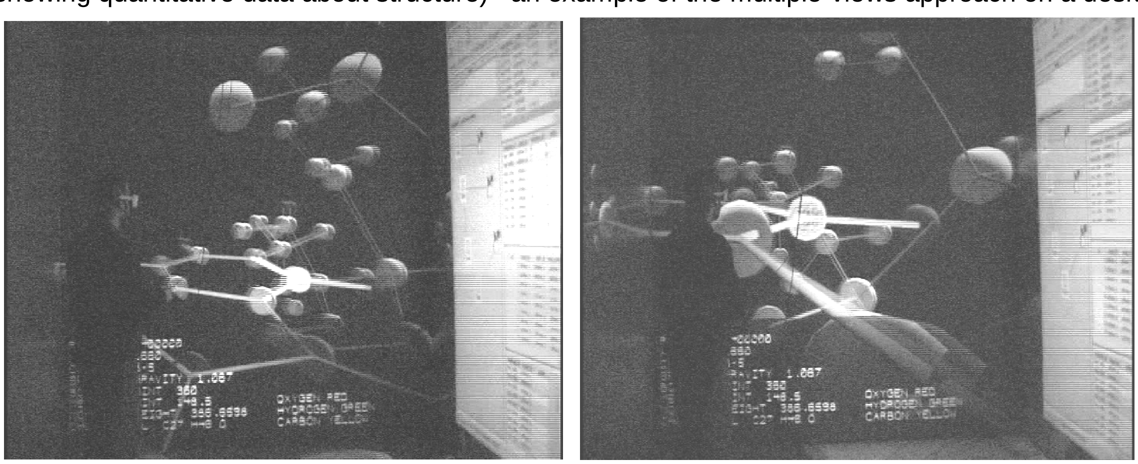

Figure 5: Relating perceptual and abstract information about molecular structures in a CAVE with multiple views (Snap2Diverse)

The specific aspects to evaluate were:

- Viability of a visualization involving simultaneous, coordinated InfoVis and VE displays
- Ability to recognize visual and interactive relationships between the views.

- User preference based on nature of data, i.e. whether users choose appropriate visualizations for different types of data.
- Effectiveness of the 3D visualization of inherently spatial data as the central basis of the IRVE.
- Use of our novel ‘XWand’ interaction system for interaction with linked visualizations in the IRVE.

Subjects for the usability evaluation were from a variety of backgrounds, including chemistry, computer science, materials science, as well as virtual reality experts from within our lab. The format of the usability study was task and response, using think-aloud protocol. The subjects were given a benchmark set of tasks to be performed. Eight tasks were formulated that can be classified into four categories: exploration tasks, search tasks, pattern recognition tasks, and comparison tasks. Exploration tasks involved loading and describing features of various chemical components. The search tasks involved getting the number of atoms or bonds in a molecule or finding a specific attribute of an atom, bond or molecule. The pattern recognition and comparison tasks asked users to detect and describe similarities and differences between two molecules such as their size, molecular weight, and shape. Users’ subjective feedback for each task was noted. We also noted down their actions such as where they search for particular information (in 2D or 3D), the problems or discomforts they face with the interaction techniques, etc.

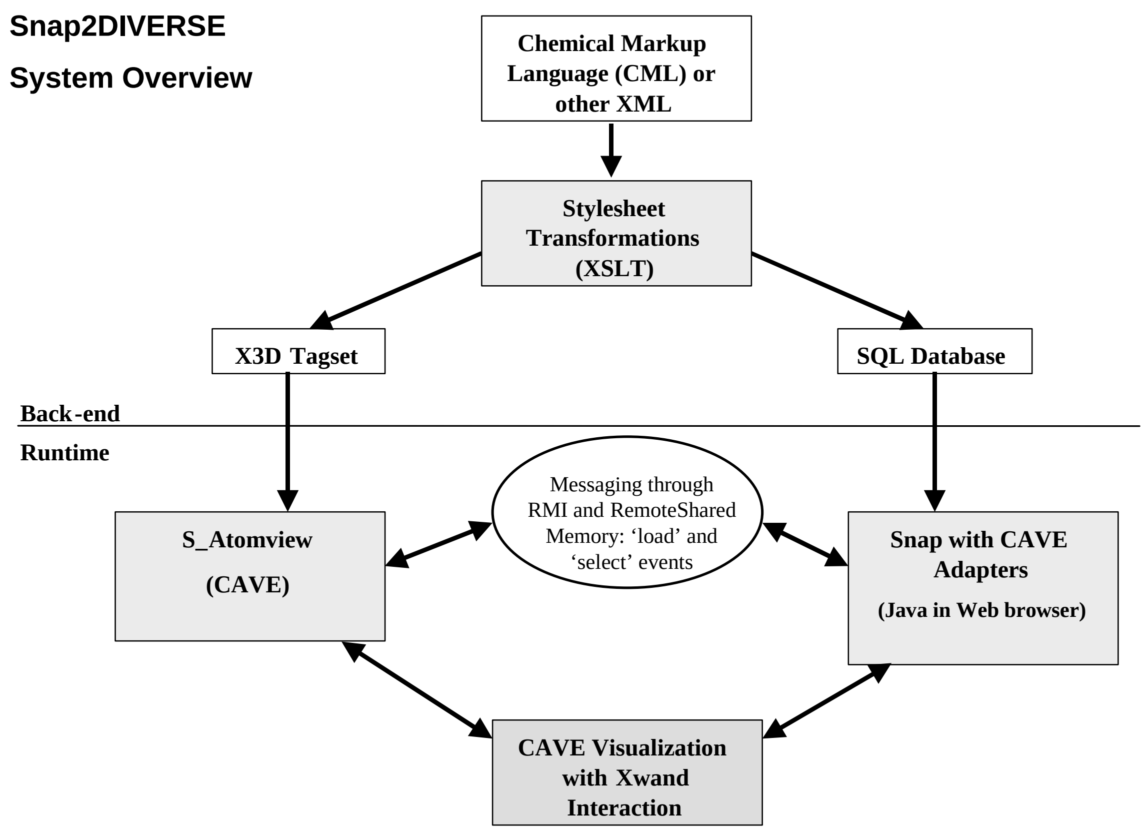

Figure 6: Snap2Diverse System Architecture (from Polys et al 2004a)

The results of the usability evaluation were obtained from user observation and the feedback questionnaire. They consist of usability measures, technical deficiencies, and suggestions by subjects. Usability measures we evaluated included: the time to understand the basic concept of coordinated 2D and 3D visualizations, the learning time for system interaction, and the degree to which a new user is overwhelmed by the CAVE.

The most important results involved the users’ interaction between the spatial and abstract information. Some tasks were designed so that users needed to answer questions about spatial properties based on some abstract information criteria or vice versa. In addition, there were tasks that could be answered by either perceptual or abstract sources. In most cases, users chose suitable visualizations to recover the information required for the finding and comparing tasks. This suggests that users were capable of interacting with and comprehending complex relationships between both the abstract and spatial information via the multiple-views IRVE design. If the task was an exploration task or pattern recognition task and could be accomplished by referring to either the perceptual or abstract information, nearly all users resorted to indexing via the spatial information. This confirms our hypothesis that the spatial information display of the IRVE would serve as a familiar grounding for users of the information space.

Learning time for the brushing interaction was surprisingly low for both VR novices and experts. The use of the wand was initially confusing for novice subjects, but after a few minutes they could fluently use the wand for: navigating by flying (thumb joystick), toggling between the navigation and selection mode (button 1), and selecting abstract information (thumb joystick and button 2). The visually implicit association between perceptual and abstract information (coordinated highlighting via selection) was established between the linked views and sufficient for completion of all the tasks by all users.

This result is important for IRVEs design as it suggests that users can operate multiple coordinated views to accomplish crucial IRVE tasks. We believe it is essential that IRVE can integrate VE softwares and InfoVis software for the qualities of parallelism of display and coordination of views. This strategy can give users easier, integrated access to information and help them to generate insight through stronger mental associations between physical and abstract information while preserving mental models of each type of information.

4.2 Biomedical Simulation: PathSim

PathSims is a computer model and simulation engine designed for Systems Biology investigators and Virologists to study the dynamics of an immune system under various infection conditions ***in silico***. Computationally-intensive simulations of anatomical structures and their infection are performed on a server and raw data results are saved in structured directories. Visualization Generator scripts read the simulation result files, compute and compose the IRVE results interface, which is also saved on the server.

Through the PathSim project, we have implemented a number of custom IRVE information and interaction objects to meet the requirements of Systems Biologists to explore multi-scale, heterogeneous information [Polys et al, 2004b]. These behavior and display objects attempt to resolve tradeoffs on the dimensions of the IRVE design space. In embedded IRVE visualizations like the current PathSim, we have been evaluating a number of design parameters including the representation, location, and association of abstract information in the VE.

For example, at a given scale, agent populations for an anatomical region can be visualized with a heatmap visualization. In this case, the abstract data is structured by the spatial data and the data values are a function of the space. This is a technique widely used in scientific visualization or visualization of population/census data. Our initial PathSim designs support a number of visual encodings such as logarithmic color scales and transparency (Figure 7). Population details for a given anatomical region can also be accessed through information panels which take the form of visual items (pop-up labels, hyperlinks). Here, the abstract data is related to localized objects in the space. We have implemented a set of IRVE behaviors encapsulated as Semantic Objects for VR scenegraphs (Figure 8).

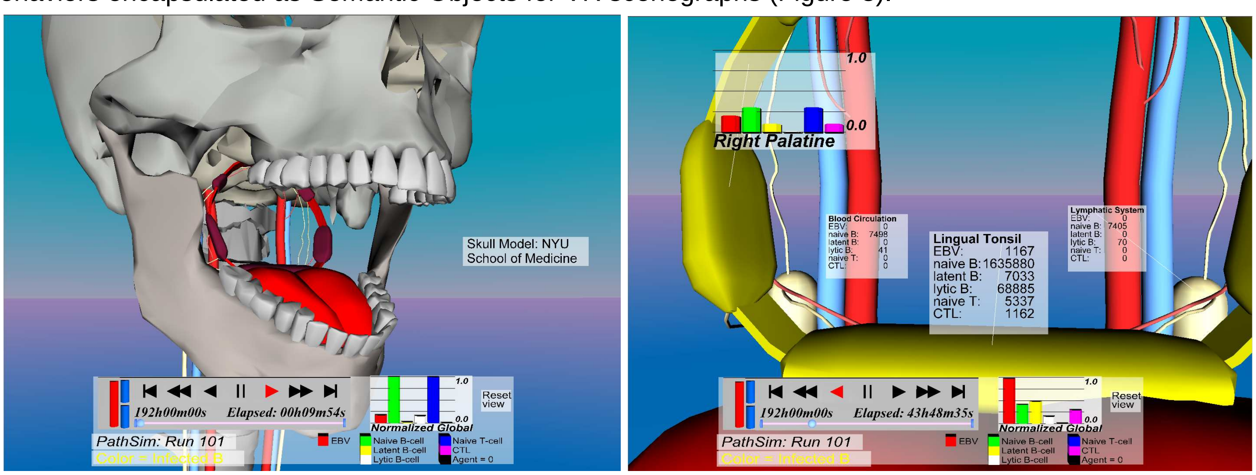

Figure 7: A prototype IRVE showing an information-rich database of macro-scale simulation results of Epstein-Barr Virus infection of the tonsils. Numbers and graphs are animated.

Semantic Objects are a conceptual and programmatic abstraction of spatial objects in the visual space of the IRVE that carry associated information along with their geometric and appearance information. Thus, information layout locations for Semantic Objects can all be described as object-fixed. The advantages of defining annotation information and display behaviors along with the objects are: display behaviors are in the scenegraph and operate independently of the display's size and resolution, and no central 'layout manager' is needed. This has made it possible to deploy Semantic Objects and their related information display objects across desktops, Head Mounted Displays, Domes, and the CAVE.

We conducted analytic usability evaluations on a number of association and location parameters in PathSim using Semantic Objects with text information panels and bar-graph panels. We used 2 common analysis techniques known as heuristic evaluation [Neilsen, 1994] and pluralistic walkthrough [Polson et al, 1992]. In heuristic evaluation, usability experts review and rank a system against generally established interface guidelines. Pluralistic walkthrough is an inspection method in which developers, users, and usability engineers collaborate to analyze a software system and identify problems. These techniques lead us to believe: automatic panel scaling (periodic), continuous orientation, and proximity-based level of detail were effective in providing details on demand functionality. We also note that nominal and categorical data are especially suited for object-fixed display with visually explicit association such as lead lines between the panel and the object.

However, two tradeoffs became apparent. The first is that with many information panels in the space, the visual field can become crowded. We mitigate this problem by rendering the information group only when the user toggle-selects the object. This feature is used to pop-up (or hide) information panels for secondary anatomical structures. The second tradeoff is that while color-coded population bar-graphs can give users a quick, qualitative understanding of the agent populations in anatomical locations, they are difficult to use for comparison. This is because in the 3D perspective, the panels may be smaller if they are more distant, and their locations in the visual field are not necessarily aligned. This is leading us to develop facilities for a user-fixed workspace where panels can be loaded and arranged. This workspace is at or near the image plane like a HUD.

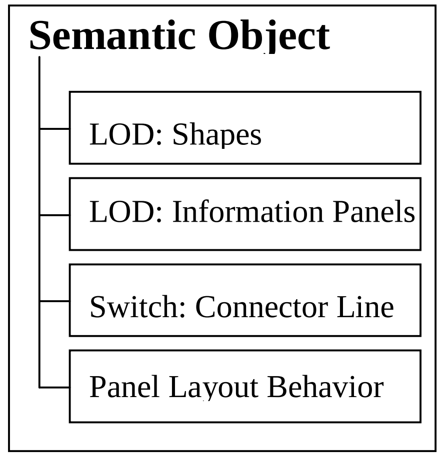

Figure 8: Encapsulating information display behaviors in the PathSim IRVE

5.0 Supporting Standards

5.2 Extensible Markup Language (XML)

The World Wide Web Consortium's (W3C) meta language codification of XML has opened new and powerful opportunities for information visualization as a host of structured data can now be transformed and/or repurposed for multiple presentation formats and interaction venues. XML is a textual format for the interchange of structured data between applications [Kay, 2001; White, 2002]. The great advantage of XML is that it provides a structured data representation built for the purpose of separating content from presentation. This allows the advantage of manipulating and transforming content independently of its display. XML also dramatically reduces development and maintenance costs by allowing easy integration of legacy data to a ***single data representation which can be presented in multiple contexts or forms*** depending on the needs of the viewer (a.k.a. the client). Data thus becomes 'portable' and different formats may be delivered and presented (styled) according to the application and user's needs.

Another important aspect of XML is the tools that it provides: the DTD and Schema. The Document Type Definitions (DTD) defines 'valid' or 'legal' document structure according to the syntax and hierarchy of its elements. The Schema specifies data types and allowable expressions for element's content and attributes; they describe the document's semantics. Using any combination of these, ***high-level markup*** tags may be defined by application developers and integration managers. This allows customized and compliant content to be built for use by authors and domain specialists. These tags could describe prototyped user-interface elements, humanoid taxonomies, or geospatial representations. Developers can describe the valid data model for their application by using the DTD and Schema, share it over the web, and standardize it amongst their community.

XML can be as strict or as open as needed: content or fragments of content can be 'well-formed' and still processed with most XML tools. Typically, data validation is at author time, but it can be done at serving, loading, or runtime if needed. From an XML-compliant source document or fragment, logical transformations (Extensible Style Sheet Transformations (XSLT)) can be applied to convert the XML data and structure to another XML document or fragment. A series of such transformations may be applied ending with a presentation-layer transformation for final delivery target style, content-type integration, and display applications [Kay, 2001; White, 2002]. Common XSLT design patterns have been described as: fill-in-the-blank, navigational, rule-based, and computational [Kay, 2001].

5.3 Extensible 3D (X3D)

The Web3D Consortium's next-generation successor to VRML is X3D. Like XML, which moves beyond just specifying a file format or a language like VRML or HTML, it is a set of objects and interfaces for interactive 3D Virtual Environments with defined bindings for multiple profiles and encodings collected under a standard API [Web3D, 2002; Walsh, 2001]. Like VRML, the X3D specification describes the abstract performance of a directed, a-cyclic scenegraph for interactive 3D worlds. In addition, it takes advantage of recent graphics advancements such a MultiTexturing and information technology advancements such as XML. X3D can be encoded with an XML binding using DTDs and Schema. The X3D Task Group has provided a DTD, Schema, an interactive editor, and a set of XSLT and conversion tools for working with X3D and VRML97. Using the XML encoding of X3D, authors can leverage all the benefits of XML and XML tools such as user-defined markup tags, XSLT, authoring environments, and server systems.

Additionally, rather than defining a monolithic standard, the X3D specification is modularized into components which make up 'Profiles'. Profiles are specific sets of functionality designed to address different applications - from simple geometry interchange or interaction for mobile devices and thin clients to the more full-blown capabilities of graphical workstations and immersive computing platforms. The notion of X3D Profiles is important for publishing visualizations and we will examine them in more detail in subsequent sections. X3D may be presented in a native X3D browser such as Xj3D, Flux, Contact and Octaga [Web3D, 2002], or transformed again and delivered to a VRML97 viewer.

X3D adds important functionality to support information-rich data models in IRVE runtime environments. For example, the X3DMetadataNode type is supported in the Core Profile and above and enables authors to include a variety of data types within the X3D scenegraph. Metadata nodes may be placed anywhere in the scenegraph; they are not rendered directly, but may have a DEF name to be accessed by Scene Access Service (SAI) Scripts. Metadata nodes provide 2 single fielded strings, which describe the standard and the name of the data field within the standard that the Metadata node represents. Metadata nodes are specified for multi-fielded integers, doubles, floats, and strings. These Metadata nodes can be aggregated under the MetadataSet node. This addition adds significant capability to X3D to describe and interchange the abstract information associated with objects and object groups.

The composition of temporal information for IRVEs is also improved in X3D. New X3DSequencerNode and X3DTriggerNode types in the Event Utilities Component provide a broad palette of functionality for authors. Composing event logics for the scenegraph runtime is now much more flexible and does not require the use of Scripts.

6.0 IRVE Outlook

6.1 Architecture

A broad variety of next-generation interface tools require that multi-scale, heterogeneous information types be integrated in a unified visualization environment. This need is most pronounced in the engineering, modeling and simulation, and design domains where spatial objects and systems interact with each other. Our engineer E had to juggle objects, attributes and scenarios across numerous applications.

In simulation for example, analysts need to understand the conditions and context of the phenomenon they are observing, especially to compare simulation outcomes. Except for configuring initial conditions, the user does not need to change the model while visualizing; they only need access to knowledgebase items in the visualization environment. Design applications in contrast, have slightly different requirements. In this case, behaviors and constraints may need to be enforced within and during visualization and the changes communicated back to the model for verification and saving.

The nature of object interactions (i.e. ontologies, behaviors, and constraints) may be represented in a knowledgebase or in the objects themselves. The latter approach has been proposed in the AEC domain as 'Smart' IFC objects [Halfawy & Froese, 2002]. While this approach does have certain advantages, the IFC schema is domain-specific and took many years of development and revision before assuming its current form. We are looking for a more general and immediate solution to support IRVEs.

In considering how our prototypes illuminate requirements for IRVEs, we have explored linked visualizations (Snap2Diverse) and embedded visualizations (PathSim). For linked visualizations, separate visualization applications (or authoring tools) may be unified through consistent object identifiers and composable event logic. This is the strategy employed in Snap and the Domos 4D Suite. In this approach, each application may use its own data model and presentation logic but must communicate through shared events. In addition, Data IDs may be equivalent or mapped between applications using a lookup table. This has the advantage of 'gluing' existing applications together and focusing on structured communication and messaging.

Embedded visualization designs such as PathSim put the abstract information within one VE application. This is achieved through both object-fixed and user-fixed display locations for temporal and abstract information (nominal, ordinal, numeric, audio, video, and hyperlinks). The display behaviors such as graphs and layout are encapsulated in the objects themselves. This independence can be an advantage and a disadvantage: while developers have complete control over the composition and coordination of abstract and temporal information within the VE, they are also responsible for all interaction and drawing functionality.

6.2 Display

In the process of implementing these design prototypes to meet our IRVE usability requirements, we have discovered deficiencies and opportunities for the evolution of X3D. In conjunction with the X3D Specification Working Group, we are developing future standards components as a foundation to address these interface requirements. These include the Annotation Component and the Compositing Component (in progress).

A proposed Annotation Component for example, would provide better support for the functionality we encapsulate in Semantic Objects. In the proposed component, associated information can be included with an object or set of objects as a multi-fielded node and rendered parallel to the display surface. In a proposed Annotation node interface, there is a reference point in 3D space (subject to the Transformation hierarchy), and an offset to the annotation information. The reference point and the annotation can be connected by a lead line of various styles. Enabling IRVE display logic does not necessarily mean to specify it. For example while proximity-based filtering of annotations and layouts of multiple panels should be supported, authors should be free to compose them as needed.

In PathSim, we defined a generic Heads-Up-Display (HUD) for user-fixed controls and abstract information. While extremely useful for maintaining the visibility of overview information and system controls, the HUD in this implementation has some drawbacks. Most importantly are the facts that the HUD is rendered with the rest of the scene and browsers vary on where they implement the near clipping plane. In cases where users have zoomed into very small scales, objects may actually come between the user and the HUD geometry. While some browsers can support ‘Overlays’ or rendering ‘Layers’, the interoperability problem can only be solved through improvement of the standard.

A proposed Compositing Component would allow more sophisticated author control over rendering (i.e. Z-order, clipping, etc). This would improve support for Heads-Up-Displays, which are common in applications, but awkward between browsers. In addition, the Compositing component could provide support for the notion of ‘ApplicationTextures’. This would provide a new level of information integration as external applications could be linked *and* embedded applications for IRVE applications.

6.3 Summary

In summary, we have developed prototype applications that attempt to address the requirements of integrated information spaces through the concept of Information-Rich Virtual Environments (IRVEs). These prototypes can help HCI researchers to investigate data model tradeoffs and the evaluate the needs for presentation techniques and target runtimes. As a result, we believe this research program will have direct impact for users such as E through experimentally validated guidelines for IRVE design and recommendations for standards architectures and functionality. Ideally, these results may lead to concise information architecture descriptions (e.g. a meta-level schema) for composing IRVE applications and displays.

The investigation of human computer interaction for information-rich 3D worlds and visualizations is still in its infancy. We expect that by enumerating effective data mappings, the combinations of coordinated information and media types, and interaction strategies for information-rich virtual environments, we can work toward advantageous computational, compositional, and convivial systems for real-time exploration, analysis, and action. This work will have a direct impact on the usability and design of such heterogeneous 3D worlds. With such mappings, coordinations, and strategies in hand, effective displays and user interfaces may be automatically generated or constructed by users depending on the expertise level and the task.

Acknowledgments

Thanks to the PathSim Team at the Virginia Bioinformatics Institute, particularly Dr. Reinhard Laubenbacher and Dr. Karen Duca for their support, insight, and guidance in the user interface design for PathSim. In addition, thanks to Dr. Ron Kriz, Director of the University Visualization and Animation Group at Virginia Tech for his support.

7.0 References

Bederson, B. B., Hollan, J.D., Perlin, K., Meyer, J., David, B., and Furnas, G (1996). "Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics." Journal of Visual Languages and Computing **7**(1): 3-32.

Bowman, D., Hodges, L., and Bolter, J. (1998). "The Virtual Venue: User-Computer Interaction in Information-Rich Virtual Environments." Presence: Teleoperators and Virtual Environments **7**(5): 478-493.

Bowman, D., D. Johnson, and L. Hodges (2001). "Testbed Evaluation of VE Interaction Techniques." Presence: Teleoperators and Virtual Environments **10**(1): 75-95.

Bowman, D. (2002). Principles for the Design of Performance-Oriented Interaction Techniques. Handbook of Virtual Environments. K. Stanney, Lawrence Erlbaum**:** 277-300.

Bowman, D., J. Gabbard, and D. Hix. (2002). "A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods." Presence: Teleoperators and Virtual Environments **11**(4).

Bowman, D., North, C., Chen, J., Polys, N., Pyla, P., and Yilmaz, U. (2003). Information-Rich Virtual Environments: Theory, Tools, and Research Agenda. Proceedings of ACM Virtual Reality Software and Technology, Osaka, Japan, ACM SIGGRAPH.

Bowman, D. W., J., Hodges, L., and Allison, D. (1999). "The Educational Value of an Information-Rich Virtual Environment." Presence: Teleoperators and Virtual Environments **8**(3): 317-331.

Card, S., J. Mackinlay, and B. Shneiderman (1999). Information Visualization: Using Vision to Think. San Francisco, Morgan Kaufmann.

Chen, J., P. Pyla, and D. Bowman (2004). Testbed Evaluation of Navigation and Text Display Techniques in an Information-Rich Virtual Environment. Virtual Reality, Chicago, IL, IEEE.

Domos Software 4D Builder Suite (2004): http://www.domos.be/site/4D/_eng/index_eng.htm

Faraday, P., and Sutcliffe, Alistair (1996). An Empirical study of Attending and Comprehending Multimedia Presentations. ACM Multimedia, Boston, MA, ACM.

Gabbard, J., D. Hix, and J. Swan (1999). "User-Centered Design and Evaluation of Virtual Environments." IEEE Computer Graphics & Applications **19**(6): 51-59.

Halfawy, M. R., and Froese, Thomas (2002). Modeling and Implementation of Smart AEC Objects: an IFC Perspective. CIB w78, Aarhus School of Architecture, International Council for Research and Innovation in Building and Construction.

Industry Alliance for Interoperability (IAI) (2002) http://www.iai-international.org

Kay, M. (2001). XSLT. Birmingham UK, Wrox Press.

Kelso, J., Arsenault, L., Kriz, R., and Satterfield, S. (2002). DIVERSE: A Framework for Building Extensible and Reconfigurable Device Independent Virtual Environments. IEEE Virtual Reality, http://diverse.sourceforge.net/.

McClean, P., Saini-Eidukat, Bernie, Schwert, Donald. Slator, Brian, and White, Alan (2001). Virtual Worlds in Large Enrollment Biology and Geology Classes Significantly Improve Authentic Learning. Selected Papers from the 12th International Conference on College Teaching and Learning (ICCTL-01). J. A. Chambers. Jacksonville, FL, Center for the Advancement of Teaching and Learning**:** 111-118.

Mine, M. (1995). Virtual Environment Interaction Techniques. Technical Report TR95-018, UNC Chapel Hill CS Dept.

Munro, A., Breaux, Robert, Patrey, Jim, Sheldon, Beth, (2002). Cognitive Aspects of Virtual Environment Design. Handbook of virtual environments: design, implementation, and applications. Mahwah, N.J., Lawrence Erlbaum Associates.

Neilsen, J. (1994). Heuristic Evaluation. Usability Inspection Methods. J. N. R. L. Mack. New York. NY, John Wiley.

North, C., and Shneiderman, B. (2000). "Snap-Together Visualization: Can Users Construct and Operate Coordinated Views?" Intl. Journal of Human-Computer Studies **53**(5): 715-739.

North, C. (2001). Multiple Views and Tight Coupling in Visualization: A Language, Taxonomy, and System. CSREA CISST Workshop of Fundamental Issues in Visualization.

North, C., Conklin, N., Idukuri, K., and Saini, V. (2002). "Visualization Schemas and a Web-based Architecture for Custom Multiple-View Visualization of Multiple-Table Databases." Information Visualization, Palgrave-Macmillan **December**.

North Dakota State University World Wide Web Instructional Committee (NDSU WWWIC) (2004) http://www.ndsu.nodak.edu/instruct/mcclean/wwwic/

Parallel Graphics Virtual Manuals (2004): http://www.parallelgraphics.com/virtual-manuals/

Polson, P. G., Lewis, C., Rieman, J., and Wharton, C. (1992). "Cognitive Walkthroughs: A method for theory-based evaluation of user interfaces." International Journal of Man-Machine Studies **36**: 741-73.

Polys, N., North, C., Bowman, D., Ray, A., Moldenhauer, M., Dandekar, C. (2004a). Snap2Diverse: Coordinating Information Visualizations and Virtual Environments. SPIE Conference on Visualization and Data Analysis (VDA), San Jose, CA.

Polys, N., Bowman, D., North, C., Laubenbacher, R., Duca, K., (2004b). PathSim Visualizer: An Information-Rich Virtual Environment for Systems Biology. Web3D Symposium, Monterey, CA, ACM Press.

Polys, N. F. (2004c). Publishing Paradigms with X3D. Information Visualization with SVG and X3D. V. G. Chanomei Chen Springer-Verlag.

Poupyrev, I., Weghorst, S., Billinghurst, M., and Ichikawa, T. (1996). The Go-Go Interaction Techniques: Non-linear Mapping for Direct Manipulation in VR. User Interface Software and Technology, ACM.

Rosson, M. B. a. C., J. (2002). Usability Engineering: Scenario Based Development of Human-Computer Interaction. New York, NY, Morgan Kauffman.

Saini-Eidukat, B., Schwert, D.P., and Slator, B.M. (1999). Designing, Building, and Assessing a Virtual World for Science Education. International Conference on Computers and Their Applications, Cancun, Mexico.

Salzman, M. C., Dede, C., Loftin, B. R., and Chen, J. (1999). "A Model for Understanding How Virtual Reality Aids Complex Conceptual Learning." Presence: Teleoperators and Virtual Environments **8**(3): 293-316.

Sheppard, L. M. (2004). "Virtual Building for Construction Projects." IEEE Computer Graphics and Applications(January/February): 6-12.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. Proceedings of IEEE Visual Languages.

Sutcliffe, A., and Faraday, P. (1994). Designing Presentation in Multimedia Interfaces. CHI, ACM Press.

Walsh, A., and Sévenier, Mikael (2001). Core Web3D. Upper Saddle River, NJ, Prentice-Hall.

White, A. R., McClean, Phillip E., and Slator, Brian M. (1999). The Virtual Cell: An Interactive, Virtual Environment for Cell Biology. World Conference on Educational Media, Hypermedia and Telecommunications (ED-MEDIA 99), Seattle, WA.

White, C. (2002). Mastering XSLT. San Francisco, Sybex.